# **Описание используемых функций**

## **HAL\_Interrupt\_Enable**

**Резюме**

**void** **HAL\_Interrupt\_Enable**( InterruptRequest\_type intRQST, **void** \*ptrHndlr )

InterruptRequest\_type intRQST – источник запроса прерываний

**void** \*ptrHndlr – указатель на функцию обработчика прерывания

**Описание**

Функция разрешает прерывание в соответствии со значением intRQST, и инициализирует адрес обработчика прерываний. (редактируется регистр IMASKL/H).

Корректные значения параметра intRQST:

*intKERNEL* = 0, *intGPIO*,

*intTMR0LP*, *intTMR1LP*,

*intUART0*, *intUART1*,

*intLINK0*, *intLINK1*,

*intNAND* = 10, *intMIL0*,

*intMIL1*, *intDIGC*,

*intDMA0*, *intDMA1*,

*intDMA2*, *intDMA3*,

*intARINC\_RX*, *intARINC\_TX*,

*intSPI1*, *intSPI2*,

*intDMA4*, *intDMA5*,

*intDMA6*, *intDMA7*,

*intI2C*, *intGTMR0*,

*intGTMR1*, *intDMA8*,

*intDMA9*, *intDMA10*,

*intDMA11*, *intUPDOWN0*,

*intUPDOWN1*, *intUPDOWN2*,

*intUPDOWN3*, *intDMA12*,

*intDMA13*, *intLCD*,

*intIRQ0* = 41, *intIRQ1*,

*intIRQ2*, *intIRQ3*,

*intSPI0*, *intSSI0*,

*intSSI1*, *intVIRPT*,

*intVCAM*, *intBUSLK*,

*intH264*, *intTMR0HP*,

*intTMR1HP*, *intALARM*,

*intTIC*, *intWDOG*,

*intHW*

**Пример**

**#include** "hal\_1967VN044.h"

**\_\_attribute**((interrupt))

**void** **DMA\_handler**(**void**)

{

//handler body

}

//Разрешение прерываний от DMA 8

**int** **main**(**void**)

{

HAL\_Interrupt\_Enable(*intDMA8*, DMA\_handler);

**return** 0;

}

## **HAL\_Interrupt\_Disable**

**Резюме**

**void** **HAL\_Interrupt\_Disable**( InterruptRequest\_type intRQST )

InterruptRequest\_type intRQST – источник запроса прерываний

**Описание**

Функция запрещает прерывания в соответствии с параметром intRQST. (редактируется регистр IMASKL/H).

**Пример**

**#include** "hal\_1967VN044.h"

//Запрет прерываний от SPI0

**int** **main**(**void**)

{

HAL\_Interrupt\_Disable(*intSPI0*);

**return** 0;

}

## **HAL\_Interrupt\_GlobalEnable**

**Резюме**

**void** **HAL\_Interrupt\_GlobalEnable**( **void** )

**Описание**

Функция глобально разрешает прерывания (устанавливает бит SQCTL\_GIE).

**Пример**

**int** **main**(**void**)

{

HAL\_Interrupt\_GlobalEnable();

**return** 0;

}

## **HAL\_Interrupt\_GlobalDisable**

**Резюме**

**void** **HAL\_Interrupt\_GlobalDisable** ( **void** )

**Описание**

Функция глобально запрещает прерывания (сбрасывает бит SQCTL\_GIE).

**Пример**

**#include** "hal\_1967VN044.h"

**int** **main**(**void**)

{

HAL\_Interrupt\_GlobalDisable();

**return** 0;

}

## **HAL\_Interrupt\_IMASKClear**

**Резюме**

**void** **HAL\_Interrupt\_IMASKClear**( **void** )

**Описание**

Функция запрещает все прерывания, сбрасывая все биты регистров IMASK L/H.

**Пример**

**#include** "hal\_1967VN044.h"

**int** **main**(**void**)

{

HAL\_Interrupt\_IMASKClear();

**return** 0;

}

## **HAL\_Interrupt\_ILATClear**

**Резюме**

**void** **HAL\_Interrupt\_ILATClear**( **void** )

**Описание**

Функция сбрасывает все запросы прерываний, сбрасывая все биты регистров ILAT L/H.

**Пример**

**#include** "hal\_1967VN044.h"

**int** **main**(**void**){

HAL\_Interrupt\_ILATClear();

**return** 0;

}

## **HAL\_Interrupt\_RqstSet**

**Резюме**

**void** **HAL\_Interrupt\_RqstSet**( InterruptRequest\_type intRQST )

InterruptRequest\_type intRQST – источник запроса прерываний

**Описание**

Функция устанавливает запрос на прерывание в соответствии с intRQST (изменяется регистр ILATL/H).

**Пример**

**#include** "hal\_1967VN044.h"

**int** **main**(**void**)

{

HAL\_Interrupt\_RqstSet(intUART0);

**return** 0;

}

## **HAL\_Interrupt\_RqstReset**

**Резюме**

**void** **HAL\_Interrupt\_RqstReset**( InterruptRequest\_type intRQST )

InterruptRequest\_type intRQST – источник запроса прерываний

**Описание**

Функция сбрасывает запрос на прерывание в соответствии с intRQST (изменяется регистр ILATL/H).

**Пример**

**#include** "hal\_1967VN044.h"

**int** **main**(**void**)

{

HAL\_Interrupt\_RqstReset(intUART1);

**return** 0;

}

# **Рекомендации для стартовой инициализации прерываний**

1. Необходимо сбросить текущее прерывание при помощи команды RDS (если процессор в режиме прерываний).
2. Очистить IMASK и ILAT.
3. Разрешить глобально прерывания.
4. Включить интересующее прерывание.

**Пример**

**#include** "hal\_1967VN044.h"

**\_\_attribute**((interrupt))

**void** **DMA\_handler**(**void**)

{

asm("nop;;");

}

//Разрешение прерываний от DMA 0

**int** **main**(**void**)

{

asm("rds;;");

HAL\_Interrupt\_IMASKClear();

HAL\_Interrupt\_ILATClear();

HAL\_Interrupt\_GlobalEnable();

HAL\_Interrupt\_Enable(*intDMA0*, DMA\_handler);

**return** 0;

}

# **Пример использования**

**#include** "hal\_1967VN044.h"

**#include** <stdint.h>

**#define** IntReq 0

**int** Cnt = 0;

**\_\_attribute**((interrupt))

**void** **Handler**()

{

Cnt++;

}

**int** **main**()

{

**int** error\_ct = 0;

HAL\_Interrupt\_ILATClear();

HAL\_Interrupt\_RqstReset(*intWDOG*);

HAL\_Interrupt\_RqstReset(*intALARM*);

HAL\_Interrupt\_RqstReset(*intTIC*);

HAL\_Interrupt\_RqstSet(*intTIC*);

HAL\_Interrupt\_GlobalEnable();

HAL\_Interrupt\_Enable(*intTIC*, &Handler);

HAL\_RTC\_InitTicPeriod(0x3FF);

LX\_RTC->RTC\_CNT = 0x0;

HAL\_RTC\_Busy();

**while**(LX\_RTC->RTC\_CNT != 2);

**if** (Cnt == 0)

error\_ct++;

**while**(1);

**return** 0;

}